A Context for Learning Programming based on Research Communities

Scheila W. Martins, António J. Mendes e António D. Figueiredo
Faculdade de Ciências e Tecnologia, Universidade de Coimbra

Abstract: This paper describes a research work that seeks to design a pedagogical strategy that may help students in initial programming learning. Special care is given to motivational issues and to assist students to maximize their learning through the conscious assessment of their self-efficacy level, while they develop a programming study behavior. The results of the two first implementations of the strategy in a real setting are presented and discussed in this article.
Index Terms: Learning programming; Research communities; Self-reflexive activity; Self-efficacy.

I. Introduction

There is a consensus among teachers and researchers that regards programming learning as a non-trivial activity, since it requires students to develop new cognitive and technical skills (Wiedenbeck, 2005). Problem solving involves cognitive processes, such as abstraction, inference and deduction, which should be supported by the students' basic literacy skills (like reading and interpreting problems) that often aren't fully developed.

The major difficulty for many students is to use the syntactic and semantic knowledge about a particular language to solve problems. Abstraction and problem solving are often underdeveloped among first year university students (Lahtinen, Ala-Mutka, & Jaarvinen, 2005). Their poor progress in this area may have several causes, including their poor literacy skills and learning strategies.

Programming learning demands commitment and effort. Often students develop a sense that they are unable to learn or to create even a very simple program, leading them to give up trying and, eventually, to dropout. In classes it is frequent to see students simply quitting when faced with some problem, just because they did not immediately see a solution, or their first attempt did not work as expected. It is important to motivate students and make them believe that they are able to learn. This is particularly important because they often do things wrong well before they start to do things right.

II. Teaching To Think

Recent changes in teaching models used in our University aimed to make students more independent and responsible for the construction of their own knowledge. Although the intentions were good, the results in programming courses have not been positive, since the dropout rates and course retaking have not improved and even increased in some cases. Students don't seem ready to assume a leading role in their learning and their motivation, self esteem and will to learn are simply not good enough. This is worrying as a good part of what we learn, either from will or need, involves motivation that must be developed through a continuous process, which includes intrinsic and extrinsic strategies to evaluate, stimulate or change the individual appetence to do or not do something.

Up to a certain extent, programming learning is similar to language acquisition by children, as it also requires a change in the way of thinking. The relationship of mutual influence between language and thought resembles the relationship between language and the programming paradigm. Thereby the "Pedagogy of Judgment" dynamic proposed by Matthew Lipman (Lipman, 1991), through his "Inquiry Community" could be used as an inspiration to the design of new didactic approaches and contexts to programming learning (Guzdial, 2009).

Lipman defends "teaching to think" using philosophical speeches, proposing teachers to readopt the Socratic teaching to K-12 students. His work shows that in many cases the lack of motivation to learn is acquired. It is more a result of non-reflective practices in the traditional classroom model, rather than an innate lack of curiosity from the students.

Although the pure philosophical speech is not *per se* a natural approach to teach programming, the methodical and reflective thought that comes from the dialogue inspired in Science Philosophy can be rather useful to help students to develop programming skills.

III. Proposed Pedagogical Strategy

The main goal is not to design a strategy that might be used unchanged in any programming learning environment. The aim is to define a set of guidelines regarding contexts, didactic activities, tools and motivational measures that may assist teachers to turn their own programming didactic practice, and inspire them in the creation of specific learning contexts for programming courses (Collins, Brown, & Holum, 1991; Biggs, 1999).

The strategy proposal was inspired by Lipman's didactic metaphor. Our research community classroom proposal includes didactic activities planned to strengthen the student's problem solving skills and their involvement with the learning process (Martins, Mendes, & Figueiredo, 2010). Collaborative tasks, like small projects, research activities, peer tutoring, and continuous assessment are good examples. The context can also include computer-based tools that might help learning, such as algorithm simulation tools or software to support competitions. To stimulate extra-class activities and to facilitate monitoring and continuous assessment tasks, it is important to use a Learning Management System (LMS).

A good in class communication is important, but sometimes it is not enough, as some students have difficulty to talk directly with the teacher, and the teacher can't listen to all students at the same time. On the other hand it was important to include in the strategy some form to promote student reflection about the course and their own learning (George, 2002). Those were the reasons to include in the strategy a biweekly reflection that students are expected to write in the course LMS.

In each reflection students have to write about what they learned in the previous two weeks, the main difficulties felt and their reasons, what they think about the course activities and rhythm, and any other aspect they feel relevant. Each reflection is only accessible to the teacher and the student who wrote it. Reading the reflections may help the teacher to identify each student specific difficulties and feelings about the course, allowing her/him to make class or individual interventions that may address the identified problems. Also having to write the reflections may force less committed students to assume that fact to the teacher, which by itself may induce some behavior change, as possibly the student wouldn't like to write that again in the next reflection.

Teachers are expected to give particular attention to student's motivation. One key issue is to make students aware that the teacher notices their efforts and improvements. For example the teacher can highlight the fact that the student is now able to solve a problem that she/he couldn't solve before. The general idea is to make students feel that their work is being observed and rewarded.

IV. Motivation and Learning

Motivation has a very important role to play in learning. Several theories and instruments have already been proposed to stimulate, maintain and measure students' motivation. Some formal tools seek to assess cognitive aspects related with the student's learning strategies and motivation.

The Inventory of Attitudes and Study Behaviors (IACHE) (Monteiro, Vasconcelos, & Almeida, 2005) assesses aspects related with higher education student's learning strategies: 1) Whether a didactic strategy satisfies a set of learning requisites in a given course; 2) Changes in student's attitudes regarding their academic performance. IACHE includes cognitive, motivational and behavioral aspects, distributed in five sub-scales: Understand focus, Reproduce focus, Involvement, Organization and Competence Personal Perception.

The Course Interest Survey (CIS) (Keller, 2009) is inspired in Keller's ARCS Motivation Model (Attention, Relevance, Confidence and Satisfaction). CIS gives information about ARCS' cognitive measures among students in a given course, such as: pedagogical approach, class rhythm, teaching practice, satisfaction with proposed activities, etc.

The Student Motivation Problem Solving Questionnaire (SMPSQ) (Margolis, McCabe, & Alber, 2004) is an informal survey that asks students to express their success and failure expectations, and how much time and energy they are willing to invest in a school task.

The self-efficacy (Bandura, 1977) is a motivation measure based on a self-analysis of the ability or inability to perform a specific task. Self-efficacy scales for programming (Ramalingam & Wiedenbeck, 1998) can help to keep students alert regarding the quality of their learning, offering them another perspective about their capacities, different from essays and exams' grades.

During the strategy implementations, the above formal instruments were used to get feedback from students, allowing a better understanding about the impact of the strategy and each of its components.

V. The strategy in action

The first implementations of the proposed strategy took place in 2008/09 and 2009/10, in a Programming course included in the Masters in Design and Multimedia (MDM). The aim of this course is that students develop basic programming abilities that may allow them to follow other courses that require programming knowledge. Although it is a master course, most students had no previous experience in programming or had some bad experiences in their previous studies in the areas of Design (Multimedia, Industrial or Communication), Arts and Architecture.

This course was chosen because one of the authors was the professor in charge of it, and the expected low number of students would make the experiment more manageable.

In other programming courses classes are usually divided in lectures and labs. In this case no distinction was made. In both years all classes were spaces for knowledge construction, often mixing different types of activities, in a total of 6 hours of work per week.

Considering the student's background the programming language used in the course was Processing, a Java like language designed to visual and artistic programming projects.

The exercises and projects proposed in the course had mostly a visual nature, but involved a need for research, especially a review of algebra and mathematics. The course used a Moodle space, where materials were available and some tasks were performed.

A. First Experiment Results

In the first experiment, called MDM1, the course had 15 students, which made possible a close student monitoring and allowed the teacher to know well each student, changing the class pace and activities whenever he felt necessary.

The following activities were used in the course: individual seminar about artistic projects developed with Processing (to raise awareness about programming artistic potentialities), small group/individual projects, code analysis (discursive evaluation of solutions created by peers), a programming based artistic portfolio, and a biweekly individual reflection.

The student's reflections were used during the course to check students' satisfaction with their own performance, tasks, materials and the pace of classes. After the course a more systematic content analysis (Bardin, 2009) of student's reflections was made, trying to find information that might help to improve the pedagogical strategy. The most common categories identified in the students writings (positively and negatively) are shown in Figure 1.

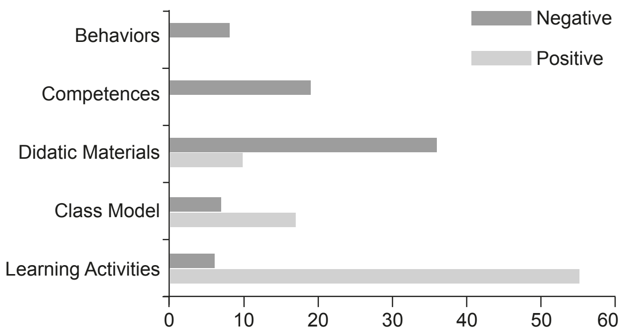

Fig. 1. Results for content analysis of MDM 1 reflections

The results showed that students were pleased with the learning activities and the class model adopted. However, they didn't like some didactic materials used, especially the number based exercises that were used in the arrays content, as well as they showed negative views about the evolution of their own competences and behaviors (commitment especially).

In the end of the course students answered an 8 question interview based on the ARCS model, in which they evaluated the course, and answered some questions related to their previous academic trajectory. The results of the interviews corroborate several aspects identified in the reflections:

- Of the five courses students attended in that semester, Programming and Internet Technologies caused more negative expectations due to frustrating experiences some students had gone through during their BSc.;

- Students associate past good experiences with courses where there was a clear relationship between didactics and the teacher's posture. They mentioned positive experiences in courses where they did not feel attracted by the themes, but felt inspired by the teacher;

- Students considered Programming a pleasant surprise, highlighting the way the course was conducted, the tools used and the evaluation process.

After this experiment some conclusions were made and some adjustments were defined. The constant monitoring proved to be very important, as it allowed a better evaluation of the learning activities impact. It also allowed the teacher to change class rhythm and activities, to address issues of commitment and dispersion, and, in particular, to help students showing signs that they might dropout from the course.

The experience made clear the need to associate a measure of motivation like self-efficacy, specifically linked with the language and the programming paradigm used, as it could help to maintain student commitment during the course.

Another decision was to introduce the cognitive tests already mentioned in second experiment. At this late stage in the first experiment only 11 students answered the IACHE survey. The data was used to help the analysis of the results obtained in the second experiment.

B. Second Experiment Results

The second experiment, called MDM2, also had 15 students (one was repeating the course). A similar context to MDM1 was used, but three new activities were added: individual programming challenges, mini-test simulation and mini-tests.

The individual programming challenges were inspired on JiTT challenges (Bailey & Forbes, 2005). Small programming tasks were used to encourage individual work, especially outside the classroom. The challenges included a self-evaluation component concerning the merit of the proposed solution, helping the students to get used to critical assessment and to reflect about their own work. The mini-test simulation was the only non-graded task. It was included to allow students to experience the mini-test situation and be aware of their level under less stressful conditions. As the results were not good, the teacher used them to raise class awareness about the need of extra work, especially outside the classroom.

This year some cognitive surveys were used to evaluate cognitive aspects related with motivation to learning: 1) IACHE to get information about students' study behavior and learning strategies; 2) CIS to measure students' motivation according to the ARCS model; 3) SMPSQ to assess the student's level of satisfaction and resistance with the different learning activities and 4) a Processing self-efficacy scale (SESP) to keep students alert regarding the quality of their learning. In all these instruments higher scores mean a better result, except in the case of IACHE's personal perception sub-scale that was designed reversely by its authors, making higher scores mean poor individual perceptions.

The small size of the samples did not allow any general statistical assumptions. The same reason prevented the use of experimental and control groups. Intensity levels analysis was used to facilitate the identification of trends in student's answer patterns. The survey's answers to each cognitive measure were grouped in three levels: low, average and high. The objective was to see if there was an improvement in the answers concentration from low to average and high levels, as an indication of the strategy impact on the student's cognitive measures. The survey's structure is presented in table I.

Table I - Test data specifications summary

	Tests				Answers per intensity level		
	I	II	Statistic Measures III		IV	V	VI
IACH	44	1 - 6	$10<\bar{x}<60$, $X_m=35$ $8<\bar{x}<48$, $X_m=28$	(1) (2)	1.2	3.4	5.6
CIS	34	1 - 5	$8<\bar{x}<48$, $X_m=28$ $9<\bar{x}<45$, $X_m=27$	(3) (4)	1,2	3	4,5
SESP	32	1 - 7	$32<\bar{x}<224$, $X_m=128$	(5)	1,2	3,4,5	6,7
SMPQS	15 5	0 - 75	$15<\bar{x}<75$, $X_m=45$ $5<\bar{x}<25$, $X\ \ =15$	(6) (7)	0,1,2	3,4	5,6

I – Number of questions in each survey, II – Answers possible in each survey, III – The range of the surveys score, minimum and maximum (Xm) and the average point (Xm), IV – Low level, V - Average level and VI – High level

It is important to note that the surveys are not a part of the strategy. They were used to get a better picture about its impact. The exception was the Processing self-efficacy scale (SESP), since its pre-test results were given to students, so that they could use them to improve their work.

In this experiment the student's cognitive aspects were checked, according to the following schedule: IACHE pre-test in September, the Processing self-efficacy scale (SESP) pre-test in October, both post-tests in February, the CIS test in November and the SMPSQ survey was filled after each activity was completed by the students. The results are presented and discussed in the next sections.

IACHE Survey Results

The statistical reference values for IACHE's understand and organization focus are given by (1), and for all other three sub-scales by (2) in table I. The averages of the post tests in both experiments and the pre-tests of MDM2 are presented in table II.

Table II - IACHE survey averages summary

PRE					POST				
I	II	III	IV	V	I	II	III	IV	V
42.6	29.9	21.5	36.6	31.4	39.9	28.1	25.0	33.1	31.4
					41.3	30.6	33.7	26.1	26.6

I-Understand, II-Reproduce, III-Personal Perception, IV- Involvement V-Organization

Comparing the averages of the two post-tests, it is possible to conclude that all values decreased from MDM1 to MDM2, except for the Involvement and Organization sub-scales. In both samples, the results of understand focus were higher than the reproduce focus, which was good, since it means that students were more concerned to understand than just to memorize contents. The Wilcoxon Signed-Rank and Mann-Whitney-U tests results are presented in table III.

Table III - Non-parametric test summary between MDM1 and MDM2

		I	II	III	IV	V
1	Negative Ranks (Pos < Pre)	7	10	2	9	8
2	Positive Ranks (Pos > Pre)	2	2	8	2	2
3	Ties (Pos = Pre)	3	0	2	1	2
4	(Wilcoxon-W) ρ-Value	.085	.134	.021	.061	.021
5	(Mann-Whitney-U) ρ-Value	.756	.654	1.00	.777	.044

I- Understand, II- Reproduce, III- Personal Perception, IV- Involvement, V- Organization

Although the samples were small and not all MDM1 students took the post-test, the results of the Mann-Whitney test (line 5 on table III) show that there were no significant behavioral differences in the two samples. The average variations between the two groups are not statistically relevant, except in the case of the organization sub-scale (ρ=0.044). This difference may have resulted from the different activities included in MDM2, which required more effort from the students, as well as a higher involvement and better time management to meet the deadlines.

Comparing the pre-test and post-test averages in MDM2 (table II) it is possible to see that all values decreased, except for personal perception. However, the Wilcoxon test (lines 1 to 4 in table III) shows that only the variations in personal perception and organization (ρ=0.021) can be regarded as statistically relevant values. If the change in organization is positive, the same cannot be said about the increase in personal perception, since a higher value in this parameter means a lower personal perception. This means that the student's level of trust in their own skills had decreased. This is very important considering the objective to reduce dropout rates. Possibly that parameter should be checked more often during the course, allowing the teacher to address the problems as early as possible with the relevant students.

The intensity level analysis results are shown in table IV.

The analysis per intensity level from MDM1 to MDM2 shows a decrease in the number of answers in the low and high levels, and an increase in the average level in MDM2. The same pattern is noticed from the pre to the post test in MDM2. The high level answers are less in MDM2 post test than in MDM1, except for organization. In general it is possible to say that MDM2 students showed more harmonious post-test cognitive indicators and less dispersion than MDM1 students.

Table IV - IACHE intensity levels summary

		PRE (%)					POST (%)				
		I	II	III	IV	V	I	II	III	IV	V
MDM 2	Low	4	20	9	3	23	5	23	10	4	31
	Av.	52	47	34	42	59	69	52	61	64	55
	High	44	33	57	55	18	26	25	29	32	14
MDM 1	Low						8	20	26	11	47
	Av.						56	41	39	48	46
	High						36	39	35	41	7

I-Understand, II-Reproduce, III-Personal Perception, IV- Involvement V-Organization

CIS Survey Results

The CIS survey was used to measure the levels of student's motivation in the middle of the course, according to the ARCS model. The statistical reference values for attention and confidence are given by (3), and for the relevance and satisfaction are given by (4), as indicated in table I. A better idea for the behavior analyzed could be obtained through the analysis of intensity levels, as mentioned before. The results are summarized in table V.

Table V - CIS survey intensity levels summary

CIS Dimension	Mean	Low (%)	Average (%)	High (%)
Attention	27.75	8	43	49
Relevance	33.00	7	32	61
Confidence	28.50	20	28	52
Satisfaction	29.92	21	31	48

The analysis shows a higher concentration of answers in the high level, which is good. However, confidence and satisfaction show a higher value in low level, which shows that some student's confidence was not high at that time. The high value of the relevance measure brings a certain balance into the results. It was possible to say that the cognitive aspects measured up to that stage were good, because the high average in relevance is a sign that students were fully aware of the importance of the course and were consciously committed.

Self-efficacy Scale Results

The scale used for Processing was translated and adapted from a scale for Java (Askar and DaVenport, 2009). It includes statements concerning the tool, the paradigm and problem solving. The statistical references values are given by (5), in table I. The pre and post test results are presented in table VI as well the intensity level analysis.

Table VI - Self-Efficacy Scale summary

PRE				PRE			
Score	Low (%)	Average (%)	High (%)	Score	Low (%)	Average (%)	High (%)
114.0	24	62	14	127.6	11	75	14

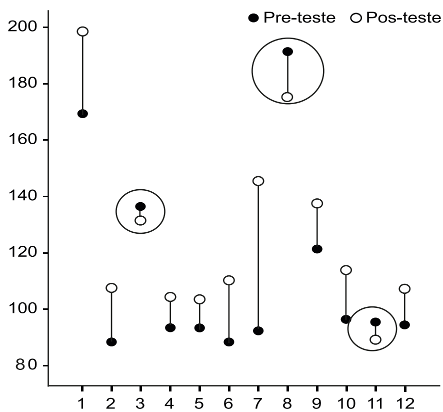

Fig. 2. Results for Processing Self-Efficacy Scale for MDM 2

The comparative analysis of averages and the drop line graphics in Figure 2 show a good evolution between the pre and post-test, even though that value is slightly below the calculated average point. The $\rho=0.021$ result in the Wilcoxon test also shows a relevant difference between both test scores.

The analysis of intensity levels revealed that the number of low level answers decreased, migrating to the average level, while the number of high level answers remained unchanged. It is possible to see that in the pre-test a little over 25% of the sample already presented individual scores higher than the average point reference value. In the post-test 75% of students increased their self-efficacy level. This is interesting when compared with the personal perception measured with IACHE, which showed worst results. That could mean that the students had better opinion about their own performance in programming than about their general ability to learn.

The SESP pre-test was given back to the students, with individual comments on their scores and explanations about the results they got. They were asked to identify which statements had low scores and then they were individually guided to establish priorities among the activities indicated by the statements. The teacher told them to focus their study efforts on increasing their trust while performing those specific tasks. Many students later stated that they used these dynamics to prepare themselves for the programming challenges and the mini-tests.

SMPSQ test Results

The SMPSQ test was used to identify the level of satisfaction and resistance students felt about the learning activities. The test is divided in two parts, the first assesses the motivation to perform a specific activity, and the second evaluates the reward expectations about the activity. Higher values in the first part are an indication that the student tends to be more motivated or less resistant to the task. The same happens in part two, where higher values mean that the student has higher expectations about the activity. The statements answered with a 0, meaning "Don't know", should be noted, as they may reveal causes for the student's resistance towards the activity and also show insecurity about their goals and their possibilities of success.

The statistical reference values for the first part are given by (6) and for the second part by (7), in table I. The intensity levels were organized as in the CIS survey, plus the Don't Know (DK) level. The results are presented in table VII.

Table VII - SMPSQ survey summary

	Mean		DK	Low	Average	High
Activities	Part 1	Part 2	(%)	(%)	(%)	(%)
Seminar	50.08	15.33	1	5	32	62
Code Analysis	49.80	17.50	3	10	41	46
Mini-test Simulation	51.07	16.69	2	8	39	51
Challenges	40.30	13.46	1	6	43	50

The statistical comparison between the activities' average did not show relevant differences. However, intensity analysis shows that the students preferred the seminar and programming challenges. It was good to see the lower concentration in low and DK levels for all activities.

Biweekly Reflection Analysis

Figure 3 shows the content analysis results of MDM2 student's reflections. The students once again evaluated very positively the class model. This time there was a specific highlight for the teacher's performance, possibly due the close individual monitoring and the good relationship established. Students also showed some evolution in their reflexive levels concerning their learning needs.

However, they showed little confidence in their own capacities as the number of negative expectations, frustrations and insecurity is high.

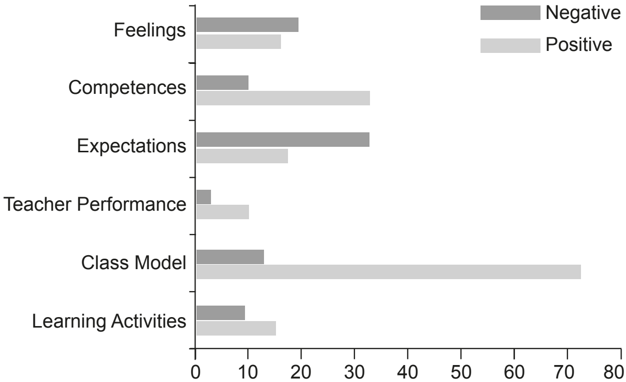

Fig. 3. Results for content analysis of MDM 2 reflections

C. Issues about Grades and Pass Rates

Both courses had high pass rates, 93.33% in MDM1 and 80% in MDM2. Pass rates and grades are important, but being able to make students realize that the efforts they make are necessary and that they can really learn was the main goal. Also, identify and minimize the conditions that may lead students to dropout was very important.

In MDM1 only one student did not pass the course, although he did not dropout. In MDM 2 three students chose not to participate in the final project, which means that they effectively dropped out. Two of them came to classes 30 days later, and therefore the course might have been harder for them. The third student dropped out of almost all disciplines.

D. The Final thoughts about the Experiments

The teacher was pleased with the strategy, not only with the results obtained, but also with class dynamics. Although there is an increase in his work, he recognizes that this approach has a good potential to promote learning. As MDM1 was the first edition of the course, there are no previous editions that might be sources of comparison.

The results obtained during the experiments were relevant to change less successful strategy aspects and to improve the teacher-student communication. This was important to allow an easier identification of students that needed a direct intervention to increase their motivation and confidence levels.

As mentioned before the small sample size doesn't allow any generalization of the results. The utilization of formal instruments aimed to give clues about student's learning and feelings about the learning activities. Another objective was to evaluate if they would be useful enough to be considered part of the strategy in the future. In this sense SESP seems the most promising, as it allows the teacher to collect information, but also to give information and directions back to students.

The reflections were meaningful in two ways. During the course, they gave some important clues to the teacher. Later, its content analysis supported or explained some statistical results. We noticed that most students shared many important emotional hints, not just about their learning progress, but also about their feelings. Most points mentioned in the reflections appeared in both experiments. Even though many students resisted to the amount of work involved in the course and, especially, to write the biweekly reflections, in the end most of them came to the conclusion that they made them stay aware of their difficulties and the level they were supposed to achieve.

VI. Conclusion

In general, the experiment received a positive evaluation both by students and the teacher. Although it was one of the most difficult courses for students, Programming received an above average classification in the students surveys carried out by the academic office. The strategy may be considered an improvement over traditional approaches for the same course context. However, some aspects pointed out on the surveys were not as positive as expected. This means further reflection is necessary and some improvements have to be considered in the next edition of the course.

The strategy was able to attract the attention of other programming teachers who have already made some changes to their own didactic approach. This strategy could be a good starting point to discuss the reform of current class models, especially for CS1, to make them more attractive and stimulating for students. As it is very different from common approaches, the proposal may contribute to a much needed debate on pedagogical practices in introductory programming courses.

References

Askar, P., & DaVenport, D. (2009). An investigation of factors related to self-efficacy for JAVA programming among engineering students. *The Turkish Online Journal of Educational Technology – TOJET,* 8 (1), Article 3.

Bailey, T., & Forbes, J. (2005). Just-in-time teaching for CS0. *SIGCSE Bull.*, 37 (1), 366-370.

Bandura, A. (1977). Self-efficacy: Toward a unifying theory of behavioral change. *Psychological Review*, 84, 191-215.

Bardin, L. (2009). *Análise de conteúdo* (4st. ed.). Lisboa: Edições 70 Ltda.

Biggs, J. (1999). What the student does: teaching for enhanced learning. *Higher Education Research & Development*, 18 (1), 57–75.

Collins, A., Brown, J. S., & Holum, A. (1991). Cognitive apprenticeship: making thinking visible. *American Educator*, 6 (1), 38–46.

George, S. E. (2002). Learning and the reflective journal. *Australian Computer Science Community*, 24 (1), 77-86.

Guzdial, M. (2009). Education: Teaching computing to everyone. *Community ACM*, 52 (5), 31-33. DOI:10.1145/1506409.1506420.

Keller, J. M. (2009). *Motivational design for learning and performance: The ARCS model approach* (1st ed.). New York: Springer.

Lahtinen, E., Ala-Mutka, K., & Jaarvinen, H.-M. (2005). *A study of difficulties of novice programmers.* In Proceedings of the 10th Annual Conference on Innovation and Technology in Computer Science Education, (New York, USA), pp. 14-18, ACM Press.

Lipman, M. (1991). *Thinking in education* (1st ed.). Cambridge: Cambridge University Press.

Margolis, H., McCabe, P. P., & Alber, S. R. (2004). Resolving struggling readers' homework difficulties: How elementary school counselors can help. *Journal of Educational and Psychological Consultation,* 15 (1), 79-110.

Martins, S.W., Mendes, A. J., & Figueiredo, A. D. (2010). *A strategy to improve student's motivation levels in programming courses.* In Proceedings of the Frontiers in Education Conference - FIE, (Washington, D.C., USA), pp. F4F-1-F4F-7, IEEE.

Monteiro, S., Vasconcelos, R. M., & Almeida, L. S. (2005*). Rendimento acadêmico: Influência dos métodos de estudos.* In Atas do VII Congresso Português de Psicopedagogia (pp. 3505-3516), Braga, Portugal.

Ramalingam, V., & Wiedenbeck, S. (1998). Development and validation of scores on a Computer Programming Self-Efficacy Scale and Group Analyses of Novice Programmer Self-Efficacy. *Journal of Educational Computing Research*, 19 (4), 365-379.

Wiedenbeck , S. (2005). Factors affecting the success of non-majors in learning to program. In *Proceedings of the First international workshop on Computing Education Research*, (New York, NY, USA), pp. 13-24, ACM Press.